

CONFIGURACIÓN DE CRON PARA AUTOMATIZAR TAREAS

En un servidor Linux, es muy importante que ciertas tareas se ejecuten en determinados momentos.

Esto se puede hacer mediante el uso de los servicios atd y cron, los que se pueden configurar para ejecutar futuras tareas.

El servicio ATD es para la ejecución de las tareas una sola vez, en cambio el servicio CRON es para tareas más recurrentes.

Dentro de un sistema Linux, algunas tareas tienen que ser automatizadas de manera regular.

Quizás sería una buena opción configurar para cada proceso una solución específica para manejar cada tarea, pero eso no sería eficiente.

Es por ello que en Linux el servicio CRON se utiliza como un servicio genérico para ejecutar determinados procesos automáticamente en ciertas horas.

El servicio CRON consta de dos componentes principales.

En primer lugar, el demonio para cron llamado CROND.

Este demonio es el encargado de ver cada minuto si hay un trabajo que hacer.

Este trabajo se define en la configuración de cron, y esta compuesto de varios archivos los cuales trabajan en conjunto para proporcionar la información adecuada para el servicio en cada momento.

El servicio cron se inicia de forma predeterminada en todos los sistemas RHEL/CentOS.

Este servicio es necesario debido a que algunas tareas del sistema se están ejecutando a través de cron.

Un ejemplo de estos es “logrotate”, un servicio que limpia los archivos de registro y se ejecuta sobre una base regular, pero otros procesos importantes de mantenimiento también se inician de forma automática a través de cron.

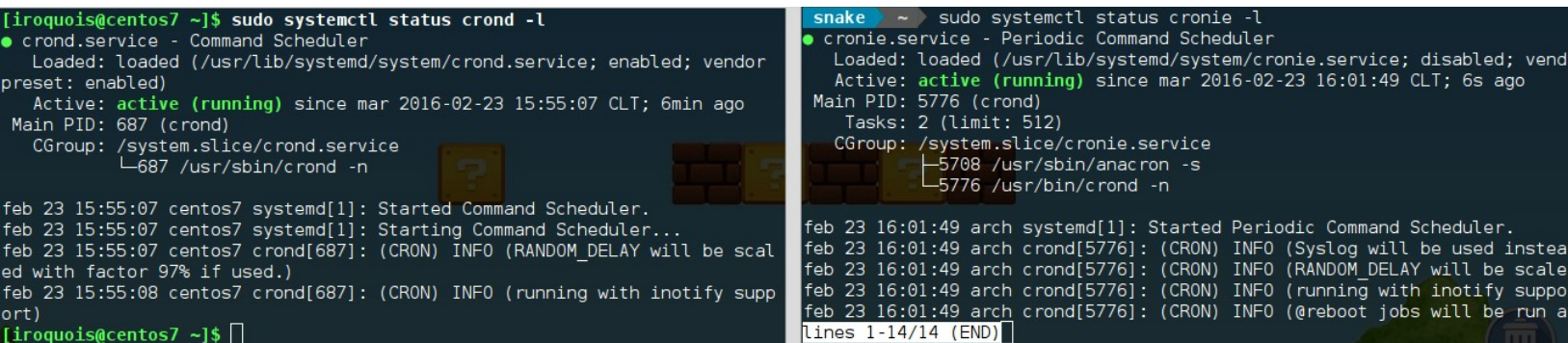
La administración del servicio CRON en sí, es fácil no se necesita mucha configuración.

Mientras que otros servicios tienen que ser reiniciados o volver a activar los cambios en su configuración, esto no es necesario para cron.

El demonio cron se despierta cada minuto y comprueba su configuración para ver si hay algo que necesita ser iniciado.

Para supervisar el estado actual del servicio cron, se puede utilizar el comando :

```
systemctl status crond -l
```



The image shows two side-by-side terminal windows. The left window is on a CentOS 7 system, showing the status of the 'crond.service' (Command Scheduler). It is loaded, enabled, and active (running) since March 2016. The right window is on an Arch Linux system, showing the status of the 'cronie.service' (Periodic Command Scheduler). It is also loaded, disabled, and active (running) since March 2016. Both outputs show the main PID and the tasks it manages.

```
[iroquois@centos7 ~]$ sudo systemctl status crond -l
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: active (running) since mar 2016-02-23 15:55:07 CLT; 6min ago
     Main PID: 687 (crond)
    CGroup: /system.slice/crond.service
            └─687 /usr/sbin/crond -n

feb 23 15:55:07 centos7 systemd[1]: Started Command Scheduler.
feb 23 15:55:07 centos7 systemd[1]: Starting Command Scheduler...
feb 23 15:55:07 centos7 crond[687]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 97% if used.)
feb 23 15:55:08 centos7 crond[687]: (CRON) INFO (running with inotify support)
[iroquois@centos7 ~]$
```

```
snake ~$ sudo systemctl status cronie -l
● cronie.service - Periodic Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/cronie.service; disabled; vendor preset: enabled)
   Active: active (running) since mar 2016-02-23 16:01:49 CLT; 6s ago
     Main PID: 5776 (crond)
    Tasks: 2 (limit: 512)
    CGroup: /system.slice/cronie.service
            └─5708 /usr/sbin/anacron -s
              └─5776 /usr/bin/crond -n

feb 23 16:01:49 arch systemd[1]: Started Periodic Command Scheduler.
feb 23 16:01:49 arch crond[5776]: (CRON) INFO (Syslog will be used instead of rsyslogd)
feb 23 16:01:49 arch crond[5776]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 97% if used.)
feb 23 16:01:49 arch crond[5776]: (CRON) INFO (running with inotify support)
feb 23 16:01:49 arch crond[5776]: (CRON) INFO (@reboot jobs will be run after 1min)
lines 1-14/14 (END)
```

La parte más importante de la salida del comando es (active (running)): donde se menciona que el servicio cron se está ejecutando.

En arch se debe instalar el paquete cronie..
`sudo pacman -S cronie`

ADMINISTRANDO EL SERVICIO CRON

El hecho de que el servicio este habilitado automáticamente (enable) significa que se inicia cada vez aunque se reinicie el sistema.

La última parte del comando muestra información de estado actual.

A través del servicio journald, el comando systemctl puede averiguar lo que está sucediendo con el servicio crond.

ENTENDIENDO EL FORMATO DE TIEMPOS EN CRON

Al programar servicios a través de cron, es necesario especificar cuándo exactamente los servicios tienen que ser iniciado.

La siguiente tabla muestra los campos de fecha y hora especificados:

Campo	Valor
minutos	0–59
horas	0–23
días del mes	1–31
meses	1–12 (o si lo prefieres puedes usar los nombres (january, march,etc)
días de la semana	0–7 (o tambien puedes usar los nombres (Sunday(domingo) es 0)

En cualquiera de estos campos, se puede usar un * para referirse a cualquier valor.

También se permiten rangos de números, como lo son las listas o patrones.

Algunos ejemplos:

* 11 * * * (Cualquier minuto entre 11:00 y 11:59)

0 11 * * 1-5 (Lunes a viernes a las 11 a.m)

*/5 2 * * 1-5 (Lunes a viernes cada 5 minutos)

Comodines!

(*) todo

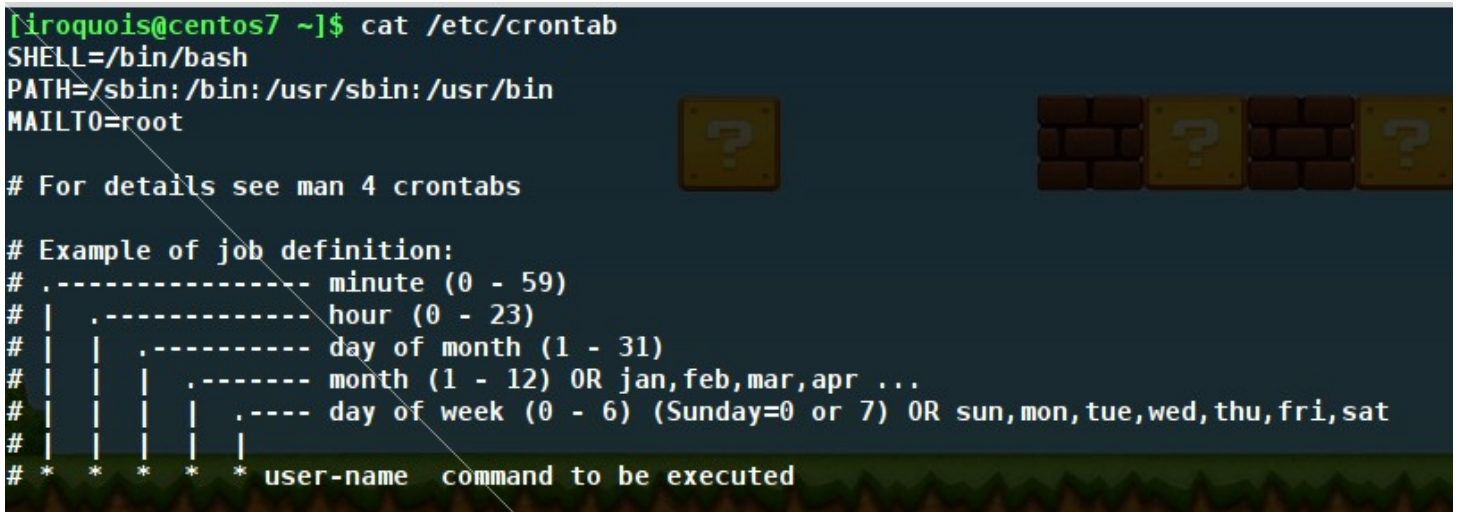
(-) rango

(,) entre

No es necesario aprender esto de memoria, si algo se olvida podemos usar el comando:

man 5 crontab

El archivo de configuración principal para cron es /etc/crontab, pero se recomienda **NO** cambiar este archivo directamente.



```
[iroquois@centos7 ~]$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan, feb, mar, apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun, mon, tue, wed, thu, fri, sat
# | * * * * *
# * * * * * user-name  command to be executed
```

En este archivo solo se da una visión general, de las diferentes especificaciones de tiempo que se pueden utilizar en cron.

También se establecen las variables de entorno que son utilizados por los comandos que se ejecutan a través de cron.

En lugar de modificar `/etc/crontab`, se utilizan diferentes archivos de configuración:

Archivos de cron `/etc/cron.d`

Scripts en `/etc/cron.hourly`, `cron.daily`, `cron.weekly`, y `cron.monthly`

Los archivos específicos del usuario se crean con `crontab -e`

Para realizar modificaciones en los trabajos de cron, hay otros lugares en los que deben especificarse los trabajos de cron.

Para empezar, los trabajos de cron se pueden crear para usuarios específicos.

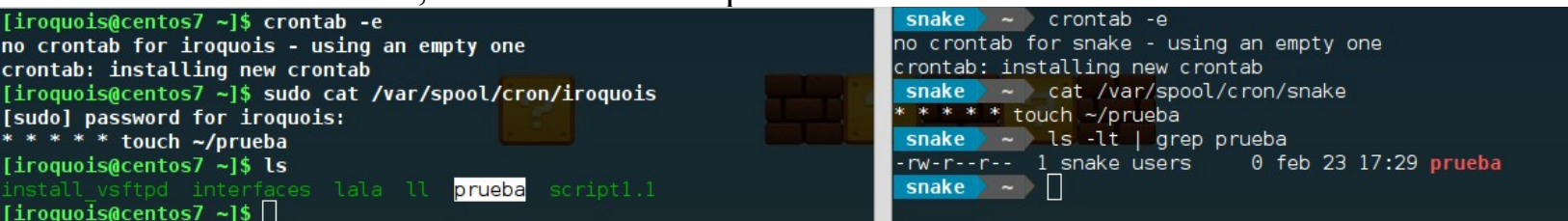
Para crear una tarea programada para un usuario, escribe `crontab -e` después de iniciar sesión, o escribe como root :

`crontab -e -u nombre de usuario`.

Cuando se está usando `crontab -e`, el editor vi(o el definido por defecto) se abre y se crea un archivo temporal.

Después de editar la configuración de cron, el archivo temporal se traslada a su ubicación final en el directorio `/var/spool/cron`

En este directorio, se crea un archivo para cada usuario.



The image shows two terminal windows side-by-side. The left window shows a user named 'iroquois' at a 'centos7' machine. They run 'crontab -e', which says 'no crontab for iroquois - using an empty one' and 'crontab: installing new crontab'. Then they run 'sudo cat /var/spool/cron/iroquois', which prompts for a password. After entering the password, they run 'touch ~/prueba' and 'ls', showing a file named 'prueba' in their home directory. The right window shows a user named 'snake' at a 'centos7' machine. They run 'crontab -e', which says 'no crontab for snake - using an empty one' and 'crontab: installing new crontab'. Then they run 'cat /var/spool/cron/snake', which shows the contents of the file: '* * * * * touch ~/prueba'. Finally, they run 'ls -lt | grep prueba', which shows the file 'prueba' with permissions '-rw-r--r--', owned by 'snake users', and dated '0 feb 23 17:29'.

Estos archivos NO deben ser editados directamente!

Cuando el archivo se guarda por `crontab -e`, se activa automáticamente.

Antes se modificaba `/etc/crontab` directamente, en la actualidad en algunos sistemas como RHEL/CentOS ya no se hace esto (adivinen en Arch tampoco)

Si desea agregar una tarea a cron que no está vinculada a una cuenta de usuario (y por esta razón por defecto serán ejecutados como root si no se especifica lo contrario), se agrega al directorio /etc/cron.d

Sólo hay que poner un archivo dentro de ese directorio (el nombre exacto no importa) y cumplir con la sintaxis correcta.

ejemplo :

```
[root@server1 cron.d]# cat unbound-anchor
# Look to see whether the DNSSEC Root key got rolled, if so check trust
and update

10 3 1 * * unbound /usr/sbin/unbound-anchor -a /var/lib/unbound/root.
anchor -c /etc/unbound/icanbndundle.pem
```

Este ejemplo contiene tres elementos:

En primer lugar se indica el tiempo, que tiene como orden de arranque a las 3:10 am en el primer día de cada mes.

A continuación, la configuración indica que el comando tiene que ser iniciado como el usuario unbound.

La última parte tiene el comando que debe iniciarse con algunos argumentos específicos y muestra cómo se debe utilizar este comando.

La última forma de programar los trabajos de cron es a través de los siguientes directorios:

- /etc/cron.hourly
- /etc/cron.daily
- /etc/cron.weekly
- /etc/cron.monthly

En estos directorios, se suelen encontrar scripts puestos por los archivos de los paquetes RPM.

Al abrir estos scripts, no se incluyen datos sobre el momento en que el comando debe ser ejecutado.

Esto es así debido a que el tiempo exacto de la ejecución en realidad no es importante. Lo único que importa es que el trabajo se puso en marcha una vez cada hora, día, semana o mes.

```
[iroquois@centos7 etc]$ sudo cat cron.daily/0yum-daily.cron
#!/bin/bash

# Only run if this flag is set. The flag is created by the yum-cron init
# script when the service is started -- this allows one to use chkconfig
and
# the standard "service stop|start" commands to enable or disable yum-cron.
if [[ ! -f /var/lock/subsys/yum-cron ]]; then
    exit 0
fi

# Action!
exec /usr/sbin/yum-cron
```

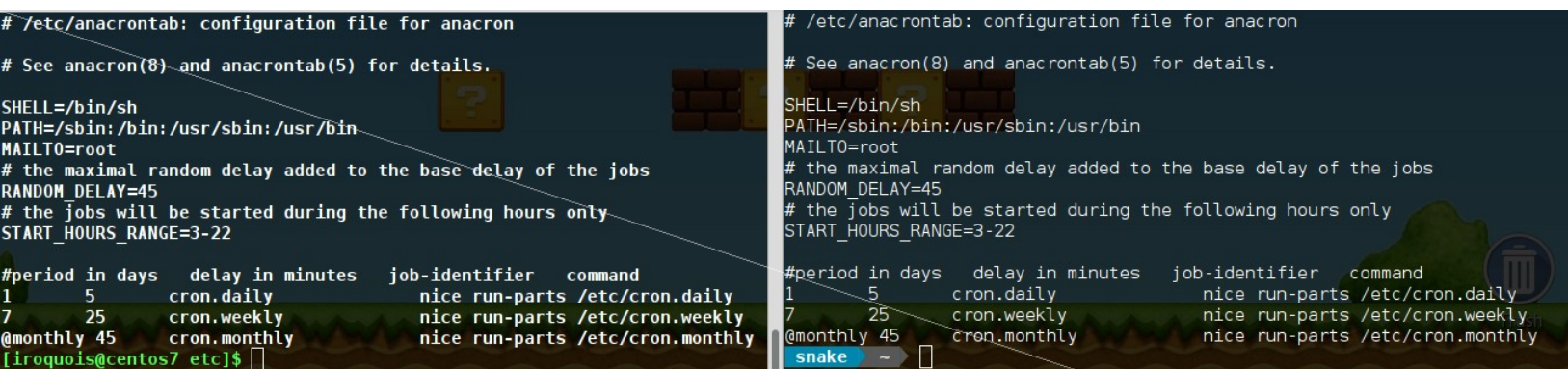

ENTENDIENDO EL PROPOSITO DE ANACRON

Para asegurar la ejecución regular del trabajo, cron utiliza el servicio anacron.

Este servicio se encarga de iniciar las tareas de cron cada hora, día, semana y mes, no importa el orden exacto.

Para determinar cómo debe hacerse, anacron utiliza el archivo `/etc/anacrontab`.

Este archivo se utiliza para especificar cómo debe ejecutar anacrontab los trabajos.



```
# /etc/anacrontab: configuration file for anacron
# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

#period in days  delay in minutes  job-identifier  command
1       5       cron.daily      nice run-parts /etc/cron.daily
7       25      cron.weekly     nice run-parts /etc/cron.weekly
@monthly 45     cron.monthly    nice run-parts /etc/cron.monthly
```

El primer campo especifica con que frecuencia se ejecuta el trabajo, expresado en días.

La segunda columna especifica el tiempo que anacron espera antes de ejecutar el trabajo, y en la última parte el comando que debe ejecutar.

TIP

Aunque es útil saber cómo funciona anacron, que por lo general no es un servicio que se configure directamente.

Realmente los servicios de anacron son llevados por los archivos `/etc/cron.hourly`, `cron.daily`, `cron.weekly`, y `cron.monthly`.

No es fácil obtener una visión general de los trabajos programados de cron. No hay un solo comando que muestre todas las tareas programadas. El comando `crontab-l` lista los trabajos de cron, pero sólo para la actual cuenta de usuario.

ADMINISTRANDO SEGURIDAD EN CRON

Por defecto todos los usuarios pueden crear trabajos.

Para denegar o permitir usuarios que los usuarios puedan crear tareas , lo podemos hacer especificando el nombre de usuario dentro de los archivos `/etc/cron.allow` y `/etc/cron.deny`

(deny = denegar)

(allow = permitir)

ATD

Considerando que cron se utiliza para programar los trabajos que deben ser ejecutados regularmente. El servicio ATD está disponible para los servicios que deben ser ejecutadas sólo una vez.

En RHEL 7, el servicio ATD está disponible por defecto, por lo que todo lo que hay que hacer es programar trabajos.

En Arch lo instalamos con :
`sudo pacman -S at`

Para ejecutar un trabajo a través del servicio de ATD, se utiliza el comando `at`, seguido por el tiempo para ser ejecutado.

Esto puede ser para un momento específico, como por ejemplo a las 14:00, pero también puede ser una indicación para una hora (apagar el sistema a determinada hora).

Después de ejecutar `(at)` se abre la shell. Para escribir el/los comandos que se ejecutarán en el momento que se especifica.

```
snake ~ systemctl status atd.service
● atd.service - ATD daemon
   Loaded: loaded (/usr/lib/systemd/system/atd.service; disabled; vendor preset: disabled)
   Active: active (running) since mar 2016-02-23 19:41:55 CLT; 20min ago
   Process: 11535 ExecStart=/usr/bin/atd (code=exited, status=0/SUCCESS)
  Main PID: 11537 (atd)
    Tasks: 1 (limit: 512)
   CGroup: /system.slice/atd.service
           └─11537 /usr/bin/atd

feb 23 19:43:00 arch atd[11580]: pam_unix(atd:session): session opened for user snake by (uid=2)
feb 23 19:43:00 arch atd[11580]: pam_unix(atd:session): session closed for user snake
feb 23 19:47:00 arch atd[11853]: pam_unix(atd:session): session opened for user snake by (uid=2)
feb 23 19:47:00 arch atd[11853]: pam_unix(atd:session): session closed for user snake
feb 23 19:50:00 arch atd[11935]: pam_unix(atd:session): session opened for user snake by (uid=2)
feb 23 19:50:00 arch atd[11935]: pam_unix(atd:session): session closed for user snake
feb 23 20:00:00 arch atd[12216]: pam_unix(atd:session): session opened for user snake by (uid=2)
feb 23 20:00:00 arch atd[12216]: pam_unix(atd:session): session closed for user snake
feb 23 20:01:00 arch atd[12288]: pam_unix(atd:session): session opened for user snake by (uid=2)
feb 23 20:01:00 arch atd[12288]: pam_unix(atd:session): session closed for user snake
snake ~ at 20:04
warning: commands will be executed using /bin/sh
at> echo "hola :v" > /home/snake/test
at> <EOT>
job 16 at Tue Feb 23 20:04:00 2016
snake ~ atq
16      Tue Feb 23 20:04:00 2016 a snake
snake ~ ls -l | grep test
-rw-r--r--  1 snake users      8 feb 23 20:04 test
snake ~ cat test
hola :v
```

Para verificar la/las tarea asignada/s :

atq

y para eliminar:

atq "numero"

Así que ya saben que hacer cuando tengan problemas en el trabajo y se termine el contrato...

